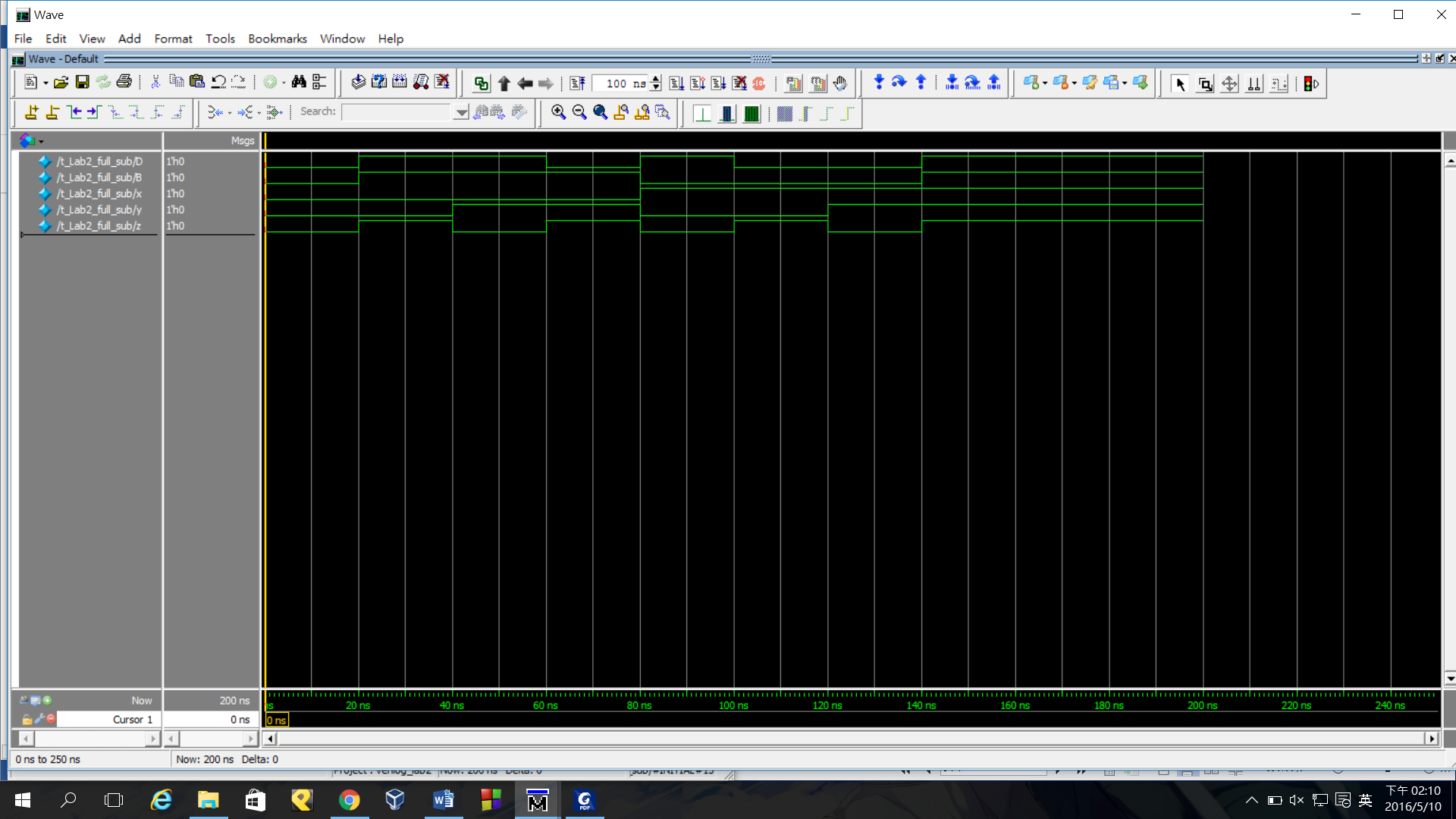


半減器的增值表：

|  |  |  |  |
| --- | --- | --- | --- |
| x | y | B | D |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 |

三種方式撰寫的半減器的波形圖皆相同，其結果也與增值表的結果相同，故正確。

(2)

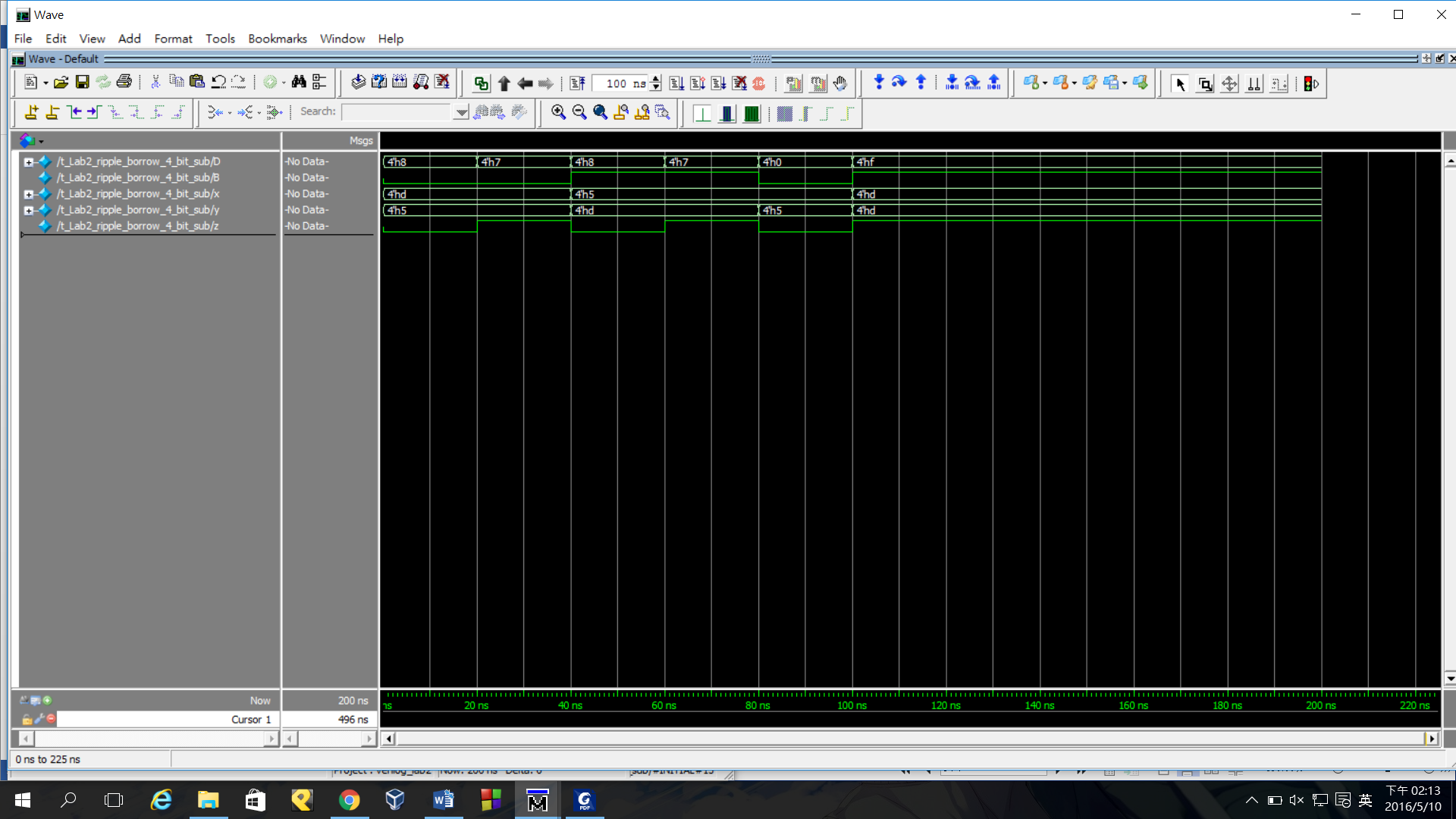


全減器的增值表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| B0 | x | y | B | D |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 |

其結果與增值表相同，故正確。

(3)

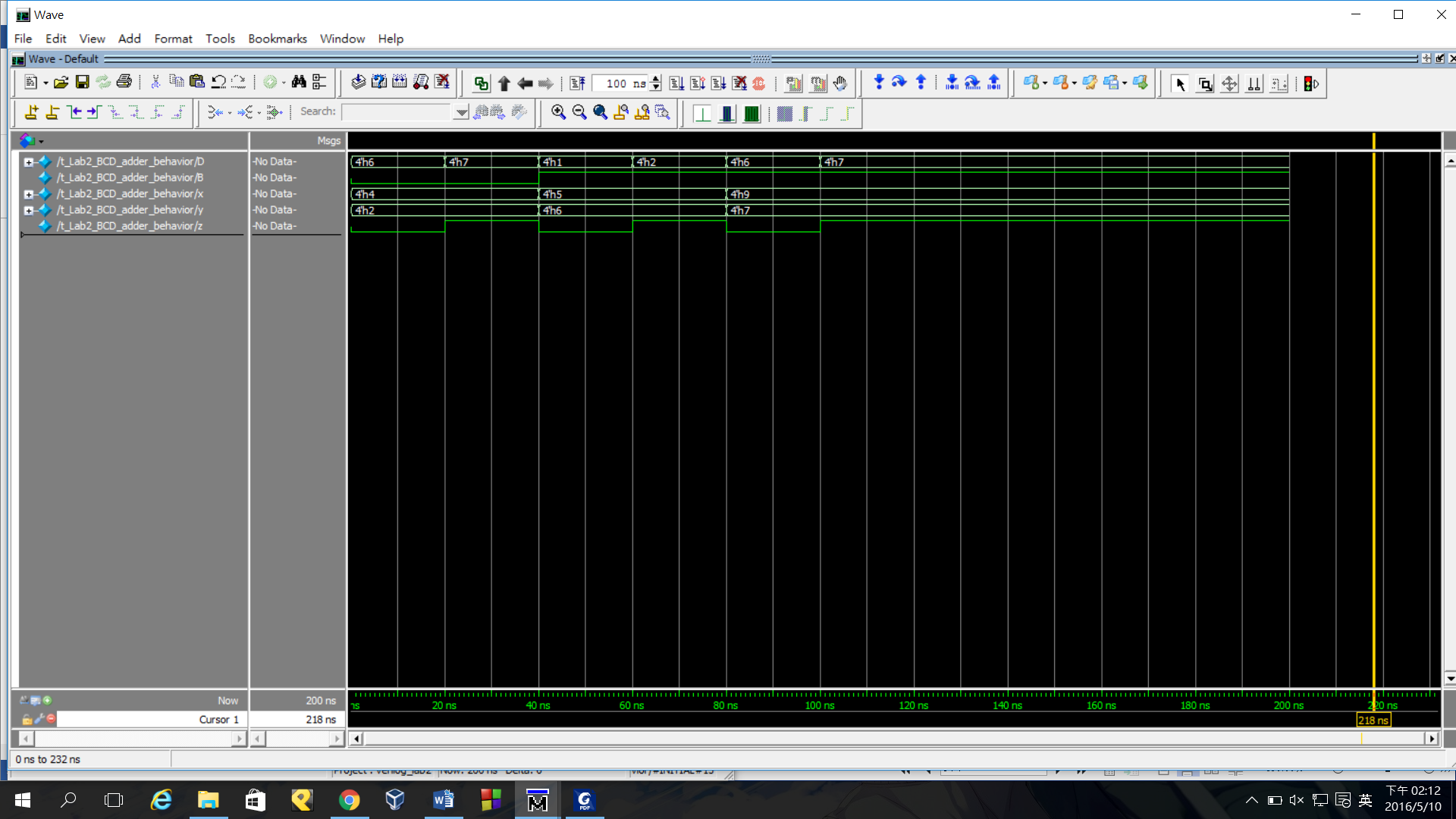


4-bit RBS的增值表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | y | Bin | Ans | Bout |
| 1101 | 0101 | 0 | 1000 | 0 |
| 1101 | 0101 | 1 | 0111 | 0 |
| 0101 | 1101 | 0 | 1000 | 1 |
| 0101 | 1101 | 1 | 0111 | 1 |
| 0101 | 0101 | 0 | 0000 | 0 |
| 1101 | 1101 | 1 | 1111 | 1 |

其結果與增值表相同，故正確。

(4)



BCD加法器的增值表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | Cin | Ans | Cout |
| 0100 | 0010 | 0 | 0110 | 0 |
| 0100 | 0010 | 1 | 0111 | 0 |
| 0101 | 0110 | 0 | 0001 | 1 |
| 0101 | 0110 | 1 | 0010 | 1 |
| 1001 | 0111 | 0 | 0110 | 1 |
| 1001 | 0111 | 1 | 0111 | 1 |

其結果與增值表相同，故正確。

(5)

這次的作業稍微比上一次的難一點。可能是因為下意識覺得寫減法器很困難吧，所以一開始一直在逃避寫它，但克服心理障礙後發現其實也沒有特別的難，寫的時間其實沒有很長，其中花比較多時間的應該是最後一題，一開始根本沒有想到其實這題那麼簡單於是就走遠了，跟朋友討論後才發現原來只是短短幾行就可以解決的程式。經過了這次的作業，我覺得我更熟悉與了解verilog的運作了，關於程式碼的規則、遇到bug時要從哪裡找出自己的bug都越來越熟悉了，相信下一次功課會更上手，讓我可以更加熟悉verilog。